

TM No. 851005



NAVAL UNDERWATER SYSTEMS CENTER
NEW LONDON LABORATORY
NEW LONDON, CONNECTICUT 06320

Technical Memorandum

PROCEDURE FOR USING THE FASP

Date: 8 January 1985

Prepared by:

Barbara Bower Welles
Barbara Bower Welles
Electronic Engineering and
Computer Applications Division
Surface Ship Sonar Department

Approved for public release; distribution unlimited.

REFERENCE COPY

| Report Documentation Page | | | | Form Approved OMB No. 0704-0188 | |
|--|------------------------------------|---|--|---|------------------------------------|
| Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. | | | | | |
| 1. REPORT DATE 08 JAN 1985 | | 2. REPORT TYPE Technical Memo | | 3. DATES COVERED 08-01-1985 to 08-01-1985 | |
| 4. TITLE AND SUBTITLE Procedure for Using the FASP | | | | 5a. CONTRACT NUMBER | |
| | | | | 5b. GRANT NUMBER | |
| | | | | 5c. PROGRAM ELEMENT NUMBER | |
| 6. AUTHOR(S) Barbara Welles | | | | 5d. PROJECT NUMBER B67800 | |
| | | | | 5e. TASK NUMBER | |
| | | | | 5f. WORK UNIT NUMBER | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Underwater Systems Center, New London, CT, 06320 | | | | 8. PERFORMING ORGANIZATION REPORT NUMBER TM No. 851005 | |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | | | | 10. SPONSOR/MONITOR'S ACRONYM(S) | |
| | | | | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) | |
| 12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited | | | | | |
| 13. SUPPLEMENTARY NOTES NUWC2015 | | | | | |
| 14. ABSTRACT This document discusses the procedure for using the FASP, Facility for Automated Software Production, located at the Naval Air Development Center in Warminster, Pennsylvania. A datalink from the VAX in building 80 to NADC exists and may be accessed to assemble SPL code which is used within the UYS-1, a signal processor utilized by each sonar system. The procedure for processing the SPL code via the FASP, transforming it into a form usable within the UYS-1, and executing the code is explained. | | | | | |
| 15. SUBJECT TERMS AN/SQR-19; FASP; Facility for Automated Software Production; UYS-1 | | | | | |
| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT Same as Report (SAR) | 18. NUMBER OF PAGES 24 | 19a. NAME OF RESPONSIBLE PERSON |
| a. REPORT unclassified | b. ABSTRACT unclassified | c. THIS PAGE unclassified | | | |

ABSTRACT

This document discusses the procedure for using the FASP, Facility for Automated Software Production, located at the Naval Air Development Center in Warminster, Pennsylvania. A datalink from the VAX in building 80 to NADC exists and may be accessed to assemble SPL code which is used within the UYS-1, a signal processor utilized by each sonar system. The procedure for processing the SPL code via the FASP, transforming it into a form usable within the UYS-1 and executing the code is explained.

ADMINISTRATIVE INFORMATION

This memorandum was prepared by Barbara Bower Welles under Project No. B67800, AN/SQR-19, Principal Investigator C. Nawrocki, Code 33B3.

The author of this memorandum is located at the New London Laboratory, Naval Underwater Systems Center, New London, Connecticut 06320.

ACKNOWLEDGEMENT

J. Vernaglia of Code 33B is responsible for providing me with the necessary documents and information which enabled me to investigate the uses of the FASP and the UYS-1.

I. INTRODUCTION

In order to use the FASP, Facility for Automated Software Production, a password and username must be obtained. Contact Charlie Cerquitella, 215-441-2486, at NADC to establish an account. The command @DBA2:[PROTEUS.19]LOGIN.COM should be placed in your login command file if the SQR-19 UYS-1 is to be accessed. Use the sonar system number that applies to you.

Note the page of references at the end of this memorandum for documents which are necessary in using the FASP and the UYS-1, for programming with SPL, and for understanding what I am about to discuss.

II. FASP PROCEDURE

Store your SPL program (example: Figure 1) and FASP job stream (example: Figure 2) in the VAX. In order to assemble, link, use the relocating loader, and encode the SPL program, the following procedure should be followed:

Access MUX. MUX enables you to use the remote job entry channel to NADC. Only one user at a time may use the datalink. In the command mode input

MUX

The system will prompt you for a site address. The proper response is

160

An input file and an output file must be opened. The input file contains your SPL program and the output file will contain whatever is returned from NADC after assembly. The commands are

I filename.type
O filename2.type

To contact NADC through the remote job entry channel use the command

REM

The datalink telephone, located behind the VAX in building 80, must now be dialed. Dial out with the TALK button pressed down, wait for a tone, press the red DATA button and hang up the receiver. The number is 93-1-215-674-9850. You will be prompted for your username and password.

To send the SPL program down to NADC use the command

GO,CR1

See Figure 3 for a list of commands which may be used on the remote job entry channel.

A short period of time will elapse and a "Ready" response will appear. To determine whether your program is ready to be returned, query the print queue by using the command

DIS,PR

Depending upon the complexity of your SPL program, the program name may not appear in the queue until hours later. The program name that appears will not be the filename you designated but a code assigned at NADC.

When your program appears in the print queue, retrieve it by using the command

```
GO,LP1
```

A "Ready" response indicates that the output file you designated contains your assembled program.

The command which will return you to MUX is

```
CTRL/Z
```

While in MUX, close the input and output files and "log keep" so you do not need to redial NADC. The appropriate commands are

```
CL 0  
CL I  
L KEEP
```

Examine your output file to insure that there are no errors. If there are, correct your file using the VAX editor, change the identification number of your program by updating the version character and date, if necessary, change ADD to REP, and begin the procedure once again. (See Figure 1).

When your program is free of errors, send the FASP job stream to NADC to generate a load module, access the relocating loader and linker, include your program in the database, and encode your program. Access the remote job entry channel as previously described and open an output file and an input file, the FASP job stream. (See Figure 2). By following the procedure explained above, the hex code of the SPL program will be returned in the output file. To exit the remote mode type a CTRL/Z and to log out of MUX type LOG. You must log out of MUX before logging off the terminal because you will remain connected to MUX if you do not.

III. VAX PROCEDURE

Hex code may be patched using the normal editing procedures on the VAX. (Note: Error messages were being produced for the use of apostrophes within the SPL code so patching was necessary.)

The hex code must be converted to binary code for the UYS-1. Using "binfile" as the name of the file to contain the binary code and "hexfile" as the name of the file containing the hex code, use the command

```
LTS binfile.type=hexfile.type
```

to achieve the hex to binary conversion.

An initial microprogram load on the UYS-1 must occur each time you wish to run a program. The command is

```
IPL
```

This sends an initializing program to the UYS-1, via the Proteus Digital Channel, which is converted to microcode and establishes the unit loader, a program on the UYS-1 that deciphers which channels are being used, and the file identification number is determined. This communication is one way from the VAX to the UYS-1. The system will respond with

```
TST
SRL
X
TST
CIW F0050000
#E0000060
#E0000020
X
$
```

if the IPL is successful. If the UYS-1's power is not on, the response will stop at CIW F0050000.

The system I/O must be invoked with the specification of a disk file which is built in with the utility task LTS. To activate the binary file in the VAX which communicates with the unit loader in the UYS-1 and establish two way communication use the command

SIO binfile.type

The system will respond with

```
SIO ADAPTER EMULATION PROGRAM
TYPE CTRL C FOR ATTENTION
```

After typing a CTRL/C, the system will respond with a question mark. A control interrupt word is being requested. Input

```
CIW 01040001
CTRL/C
```

The system will respond with another question mark, requesting the data word

```
DW 00056000
```

to indicate UYS-1 source channel 5 and file group 6 are being accessed. The system will respond with

```
#01800000
CHANNEL RESET
#01000000
```

To exit SIO input

```
CTRL/C
X
```

TM No. 851005

Your applications program stored in the VAX (example: Figure 4) which interacts with the SPL program loaded into the UYS-1 may now be executed.

To debug the SPL program, use the Test Bed Command Language in order to examine the contents of storage locations within the UYS-1. Refer to the user's manual for more information.

```

JOHN,CM45000,T100,P2.
ACCOUNT,CD1880,VERA511.
GET,FASP/UN=SC3024.
FASP.
**EOR**
L F=ASP,U=NUSC53C,K=MS
MODSW ID=DM,DBV=CURRENT,PURGE=NO,EXTLDMD=NULL,LS=1,
XLAT=ASSEM,SAVLST=NO
**EOR**
SEP114ABW
ADD      RDTM2
M
I
RDTM2    CSECT
R0       EQU 0
R1       EQU 1
R2       EQU 2
R3       EQU 3
R6       EQU 6
R7       EQU 7
START    BALR R2,0
          USING NEXT,R2
NEXT     LA R3,CHANI
          LA R0,CHANO
TAGONE   TCR R3
          BNC 1,TAGONE
          LA R1,80
          NIMA R1,8
          BC 8,TAGONE
          LA R7,84
          LA R1,85
          LIMA R6,0
          MIMS 0,164
          STA R7,SZE
          SCLK R6
TAGBLK   XIO R3,IOC1
TAGTWO   TCR R3
          BNC 1,TAGTWO
          BCT R1,TAGBLK
          RCLK R6
          STA R6,BUF3
          XIO R0,IOC2
TAGTHR   TCR R0
          BNC 1,TAGTHR
          XIO R0,IOC3
TAGFR    TCR R0
          BNC 1,TAGFR
          B ,TAGONE
          CNOP
IOC1     DS 1H
SZE      DS 1H
          DC A(BUF1)
IOC2     DS 1F
          DS 1F
IOC3     DS 1F
          DC A(BUF2)
BUF2     DS 1H
BUF3     DS 1H
CHANI    DS 1H

```

Figure 1. Example SPL Program

TM No. 851005.

```
CHANO    DS 1H
BUF1     DS 4000F
          END
**EOR**
**EOF**
```

Figure 1 (continued)

```
JOHN,CM45000,T100,P2.
ACCOUNT,CD1880,VERA511.
GET,FASP/UN=SC3024.
FASP.
**EOR**
L F=ASP,U=NUSC53C,K=MS
GENLDMD ID=DM,UN=CD1880,DBV=CURRENT,PFNAME=LOADTP,EXTLDMD=NULL,LDDIR=INPUT,
    PLAT=NAVY
**EOR**
LINKEROFF
**EOR**
PHASE TIO,STORAGE=CP,2048
HEADER FG=6,FN=2,RN=1
INCLUDE RDTM2
EOF
EOF
**EOR**
ENCODTP VSN=NA,FORMAT=OLD,PFNAME=TEMP2,LDNAME=#NULL,LDFILE=LOADTP
SENDTP PFNAME=TEMP2
**EOR**
**EOF**
```

Figure 2. Example FASP Job Stream

RBF AND EXPORT COMMAND COMPARISON

| EXPORT COMMAND | RBF COMMAND(s) |
|-----------------------------|----------------------------|
| (A)GAIN | (REW)IND,LP1 |
| (A)GAIN,nnn | SKIP,LP1,-nnn |
| (B)UP | ----- |
| (C)ONTINU | GO,LP1 |
| (D)ROP,jn | (PUR)GE,JOB=jn |
| (E)ND,CR | ABORT,CR1 |
| (E)ND,LP | SKIP,LP1,DFL |
| (F)ORWARD,pages | SKIP,LP1,nnn |
| (G)O | GO,CR1 |
| (I)GNORE | GO,CR1 or LP1 |
| (K)ILL,jn | (PUR)GE,JOB=jn |
| (L)OGIN,userid,passwd | LOGIN |
| LOGOUT | LOGOUT or LOGOFF |
| (M)ESSAGE | ----- |
| (O)UTPUT,jn,ti | (DIV)ERT,JOB=jn,USR=userid |
| (O)UTPUT,jn | (DIV)ERT,JOB=jn |
| (P)RIOR,jn,new | CHANGE,JOB=jn,PRI=new |
| (Q)UEUE,A | ----- |
| (Q)UEUE,E | (DIS)PLAY,EX |
| (Q)UEUE,I | (DIS)PLAY,IN |
| (Q)UEUE,O | (DIS)PLAY,PR |
| (Q)UEUE,R | ----- |
| (Q)UEUE,S | ----- |
| (Q)UEUE,T | ----- |
| (R)EAD | GO,CR1 |
| (R)EAD,xxx | (RES)TORE,CR1,ACK |
| | GO,CR1 |
| (S)USPEND | STOP,LP1 |
| (S)USPEND,xxx | STOP,LP1 |
| | (RET)URN,LP1 |
| | or STOP,LP1,END |
| (T)ALLY,jn | ----- |

Where

jn = job name (computer assigned)
 new = new job priority (0 <= new <= 7777)
 nnn = times 8 sectors (1 sector = 640 characters)
 pages = page count
 xxx = any non-null character

Figure 3. Remote Job Entry Commands

```

PROGRAM PDC2
C   SEND DATA FROM VAX TO UYS-1 VIA PROTEUS DIGITAL CHANNEL
C   AND DISPLAY THE TIME IT TOOK TO DO THIS
C*****VARIABLES*****
C*   OCIWN   OUTPUTTED CONTROL INTERRUPT WORD
C*           UPPER 16 BITS CONTAIN BLOCK SIZE, LOWER 16 CONTAIN NUMBER OF
C*           BLOCKS TO BE TRANSFERRED
C*   OCIW    (ORIGINAL) OUTPUTTED CONTROL INTERRUPT WORD, CONTAINS BLOCK
C*           COUNT
C*   ICIW    INPUTTED CONTROL INTERRUPT WORD(=TIME IT TOOK TO TRANSFER
C*           BLOCKS, 16 LOW ORDER BITS OF CLOCK)
C*   ICIW1   SAME AS ABOVE EXCEPT IS 32 HIGH ORDER BITS
C*   HIGHO   CONTAINS TIME IT TOOK TO TRANSFER DATA
C*   ISTAT   STATUS OF QIO WHEN INPUTTING (INTO VAX)
C*   OSTAT   STATUS OF QIO WHEN OUTPUTTING (FROM VAX)
C*   CHANI   CONTAINS DEVICE NUMBER OF INPUT CHANNEL(SOURCE)
C*   CHANO   CONTAINS DEVICE NUMBER OF OUTPUT CHANNEL(SINK)
C*   TAPE    CONTAINS DEVICE NUMBER OF TAPE-VAX CHANNEL
C*   EVOD    CONTAINS A ZERO OR ONE DEPENDING ON WHETHER JMOD'S ARGUMENT IS
C*           EVEN OR ODD
C*   W       STATUS OF WAITFR FUNCTION
C*   X       COUNTER USED AS JMOD'S ARGUMENT
C*   Y       COUNTER USED TO FILL TAPE WITH DUMMY INFO
C*   TEMP    USED TO TEMPORARILY CONTAIN OCIW (BLOCK COUNT) WHEN FILLING TAPE
C*   I       COUNTER USED AS SUBSCRIPT OF ARRAY BUF
C*   SIZE    CONTAINS SIZE OF BLOCKS TO BE TRANSFERRED
C*   NUM     CONTAINS NUMBER OF BYTES TO BE TRANSFERRED
C*   ANS     CONTAINS USER'S ANSWER TO QUESTION OF WHETHER TO FILL TAPE
C*   BLK1    AN ARRAY CONTAINING DATA TO BE TRANSFERRED TO THE UYS-1, MAX
C*           4000 32 BIT WORDS/BLOCK
C*   BLK2    SAME AS BLK1
C*   BUF     SAME AS BLK1
C*****FUNCTIONS*****
C*   SYSS$ASSIGN PROVIDES A PROCESS WITH AN I/O CHANNEL SO I/O OPERATIONS
C*               MAY BE PERFORMED ON A DEVICE, ASSIGNS A DEVICE NAME TO
C*               A LOCATION
C*   PARAMETERS:  DEVICE NAME
C*               ADDRESS OF WORD TO RECEIVE CHANNEL NUMBER
C*               ASSIGNED
C*   SYSS$QIO     INITIATES AN I/O OPERATION BY QUEUEING A REQUEST TO
C*               A CHANNEL ASSOCIATED WITH A SPECIFIC DEVICE
C*   PARAMETERS:  NUMBER OF EVENT FLAG TO BE SET AT COMPLETION
C*               NUMBER OF I/O CHANNEL ASSIGNED TO REQUESTED
C*               DEVICE
C*               FUNCTION CODE AND MODIFIER BITS SPECIFYING THE
C*               OPERATION TO BE PERFORMED
C*               BUFFER ADDRESS
C*               BYTE COUNT
C*   SYSS$WAITFR  WAIT FOR A SINGLE EVENT
C*               TESTS A SPECIFIC EVENT FLAG AND RETURNS WHEN FLAG IS SET
C*   PARAMETER:   NUMBER OF EVENT FLAG FOR WHICH TO WAIT
C*   SYSS$QIOW    QUEUE I/O REQUEST AND WAIT FOR EVENT FLAG
C*               COMBINES QIO AND WAITFR FUNCTIONS
C*   PARAMETERS:  SAME AS QIO
C*   JMOD         RETURNS THE REMAINDER WHEN THE FIRST ARGUMENT
C*               IS DIVIDED BY THE SECOND
C*****
C   INTEGER*4 OCIW,SYSS$ASSIGN,SYSS$QIOW,SYSS$WAITFR,SYSS$QIO,OCIWN,ANS,
C   ISTAT,OSTAT,CHANI,CHANO,W,X,EVOD,TAPE,I,Y,TEMP,ICIW1,ICIW,NUM,SIZE,

```

Figure 4. Example VAX Applications Program

```

1BLK1(4000),BLK2(4000),BUF(4000)
REAL*8 HIGHO
C    ASSIGN LOGICAL UNIT NUMBERS TO PDC I/O CHANNELS
    ISTAT=SYSS$ASSIGN('PDCIN',CHANI,,)
    IF (ISTAT.NE.1) GO TO 230
    OSTAT=SYSS$ASSIGN('PDCOUT',CHANO,,)
    IF (OSTAT.NE.1) GO TO 270
C    ASSIGN LOGICAL UNIT NUMBER TO CHANNEL OF MAGNETIC TAPE
    ISTAT=SYSS$ASSIGN('_MTA0',TAPE,,)
    IF (ISTAT.NE.1) GO TO 230
90   WRITE (6,100)
100  FORMAT(1X,'INPUT NUMBER OF BLOCKS TO BE TRANSFERRED'//,
1X,'(UP TO 9999)')
    READ (6,110) OCIW
110  FORMAT(I4)
    IF (OCIW.EQ.0) GO TO 290
114  WRITE(6,115)
115  FORMAT(1X,'INPUT BLOCK SIZE(BETWEEN 4 AND 4000)'//,
1X,'THIS WILL BE THE NUMBER OF 32 BIT WORDS IN ONE BLOCK')
    READ(6,117) SIZE
117  FORMAT(I4)
    IF (SIZE.GT.4000) GO TO 114
    IF (SIZE.LT.4) GO TO 114
    NUM=4*SIZE
C    NUMBER OF BYTES COMPUTED
120  WRITE (6,130)
130  FORMAT(1X,'DO YOU WANT THE REQUESTED NUMBER OF BLOCKS'//,
1X,'WRITTEN TO MAGNETIC TAPE?'//,
1X,'(TYPE A 1 FOR YES OR A 0 FOR NO)')
    READ (6,150) ANS
150  FORMAT(I1)
    IF (ANS.EQ.1) GO TO 160
    IF (ANS.EQ.0) GO TO 180
    GO TO 120
C    FILL TAPE WITH DUMMY INFO
160  Y=0
    TEMP=OCIW+1
170  Y=Y+1
    DO I=1,SIZE
    BUF(I)=Y
    END DO
    OSTAT=SYSS$QIOW(,%VAL(TAPE),%VAL('20'X),,,,BUF,%VAL(NUM),,,,)
    IF (OSTAT.NE.1) GO TO 270
    TEMP=TEMP-1
    IF (TEMP.NE.0) GO TO 170
    OSTAT=SYSS$QIOW(,%VAL(TAPE),%VAL('24'X),,,,,,,,,)
    IF (OSTAT.NE.1) GO TO 270
C    TAPE IS REWOUND
C    SEND A CONTROL INTERRUPT WORD
180  OCIWN=(SIZE*65536+OCIW)
    OSTAT=SYSS$QIOW(,%VAL(CHANO),%VAL('20'X),,,,OCIWN,%VAL(4),,,,)
    IF (OSTAT.NE.1) GO TO 270
    X=2
C    FILL BLOCK1
    ISTAT=SYSS$QIOW(,%VAL(TAPE),%VAL('21'X),,,,BLK1,%VAL(NUM),,,,)
    IF (ISTAT.NE.1) GO TO 230
C    BEGIN TO SEND BLOCK1
    OSTAT=SYSS$QIO(%VAL(1),%VAL(CHANO),%VAL('320'X),,,,BLK1,%VAL(NUM),,,,)
    IF (OSTAT.NE.1) GO TO 270

```

```

C      BEGIN TO FILL BLOCK2
      ISTAT=SYSSQIO(%VAL(2),%VAL(TAPE),%VAL('21'X),,,,BLK2,%VAL(NUM),,,, )
      IF (ISTAT.NE.1) GO TO 230
190    OCIW=OCIW-1
      IF (OCIW.EQ.0) GO TO 210
      EVOD=JMOD(X,2)
      X=X+1
      IF (EVOD.EQ.0) GO TO 200
C      WAIT FOR BLOCK1 TO FINISH BEING FILLED
      W=SYSSWAITFR(%VAL(1))
      IF (W.NE.1) GO TO 250
C      WAIT FOR BLOCK2 TO FINISH BEING SENT
      W=SYSSWAITFR(%VAL(2))
      IF (W.NE.1) GO TO 250
C      BEGIN TO SEND BLOCK1
      OSTAT=SYSSQIO(%VAL(1),%VAL(CHANO),%VAL('320'X),,,,BLK1,%VAL(NUM),,,, )
      IF (OSTAT.NE.1) GO TO 270
C      BEGIN TO FILL BLOCK2
      ISTAT=SYSSQIO(%VAL(2),%VAL(TAPE),%VAL('21'X),,,,BLK2,%VAL(NUM),,,, )
      IF (ISTAT.NE.1) GO TO 230
      GO TO 190
C      WAIT FOR BLOCK1 TO FINISH BEING SENT
200    W=SYSSWAITFR(%VAL(1))
      IF (W.NE.1) GO TO 250
C      WAIT FOR BLOCK2 TO FINISH FILLING
      W=SYSSWAITFR(%VAL(2))
      IF (W.NE.1) GO TO 250
C      BEGIN TO FILL BLOCK1
      ISTAT=SYSSQIO(%VAL(1),%VAL(TAPE),%VAL('21'X),,,,BLK1,%VAL(NUM),,,, )
      IF (ISTAT.NE.1) GO TO 230
C      BEGIN TO SEND BLOCK2
      OSTAT=SYSSQIO(%VAL(2),%VAL(CHANO),%VAL('320'X),,,,BLK2,%VAL(NUM),,,, )
      IF (OSTAT.NE.1) GO TO 270
      GO TO 190
210    W=SYSSWAITFR(%VAL(1))
      IF (W.NE.1) GO TO 250
      W=SYSSWAITFR(%VAL(2))
      IF (W.NE.1) GO TO 250
C      READ IN TIME
      ISTAT=SYSSQIOW(%VAL(CHANI),%VAL('321'X),,,,ICIW1,%VAL(4),,,, )
      IF (ISTAT.NE.1) GO TO 230
C      HIGH ORDER BITS OF CLOCK READ IN
      ISTAT=SYSSQIOW(%VAL(CHANI),%VAL('321'X),,,,ICIW,%VAL(4),,,, )
      IF (ISTAT.NE.1) GO TO 230
C      LOW ORDER BITS READ IN
      HIGHO=(ICIW1*65536.)
      HIGHO=HIGHO+ICIW
C      HIGH ORDER 32 BITS OF CLOCK(TIMES 2 TO THE 16)ADDED TO LOW ORDER 16 BITS
      HIGHO=(HIGHO/1000000.)
C      CONVERT FROM MICROSECONDS TO SECONDS
      WRITE(6,220)
220    FORMAT(1X,'THE TIME IT TOOK THE REQUESTED BLOCKS TO BE',//,
      1X,'TRANSFERRED FROM THE VAX TO THE UYS-1 IS')
      WRITE(6,221) HIGHO
221    FORMAT(F18.12,' SECONDS',/,/,/,/)
      OSTAT=SYSSQIOW(%VAL(TAPE),%VAL('24'X),,,,,,,,,)
      IF (OSTAT.NE.1) GO TO 270
C      TAPE IS REWOUND
      GO TO 90

```

Figure 4 (continued)

```
230     WRITE (6,240) ISTAT
240     FORMAT(' STATUS OF INPUT CHANNEL=',Z4)
      GO TO 290
250     WRITE (6,260) W
260     FORMAT(' STATUS OF WAIT FOR=',Z4)
      GO TO 290
270     WRITE (6,280) OSTAT
280     FORMAT(' STATUS OF OUTPUT CHANNEL=',Z4)
290     END
```

APPENDIX A

Appendix A contains all the possible VAX/VMS System I/O function codes and modifiers to these codes for VAX/UYS-1 interaction via the Proteus Digital Channel. Note reference 9 and Figure 4 for information on using these.

VAX/VMS I/O Function Codes

| | |
|------------------|----|
| IO\$_ACCESS | 32 |
| IO\$_ACPCONTROL | 38 |
| IO\$_AVAILABLE | 11 |
| IO\$_CLEAN | 1E |
| IO\$_CONINTREAD | 3C |
| IO\$_CONINTWRITE | 3D |
| IO\$_CREATE | 33 |
| IO\$_DEACCESS | 34 |
| IO\$_DELETE | 35 |
| IO\$_DIAGNOSE | 1D |
| IO\$_DRVCLR | 04 |
| IO\$_DSE | 15 |
| IO\$_ENDRU1 | 3A |
| IO\$_ENDRU2 | 3B |
| IO\$_ERASETAPE | 06 |
| IO\$_FORCE | 37 |
| IO\$_FORMAT | 1E |
| IO\$_INITIALIZE | 04 |
| IO\$_LOADMCODE | 01 |
| IO\$_LOGICAL | 2F |
| IO\$_MODIFY | 36 |
| IO\$_MOUNT | 39 |
| IO\$_NETCONTROL | 36 |
| IO\$_NOP | 00 |
| IO\$_OFFSET | 06 |
| IO\$_PACKACK | 08 |
| IO\$_PHYSICAL | 1F |
| IO\$_QSTOP | 07 |
| IO\$_RDSTATS | 0D |
| IO\$_READHEAD | 0E |
| IO\$_READINIT | 3C |
| IO\$_READLBLK | 21 |
| IO\$_READPBLK | 0C |
| IO\$_READPRESET | 19 |
| IO\$_READPROMPT | 37 |
| IO\$_READTRACKD | 10 |
| IO\$_READVBLK | 31 |
| IO\$_RECAL | 03 |
| IO\$_RELEASE | 05 |
| IO\$_REREADN | 16 |
| IO\$_REREADP | 17 |
| IO\$_RETCENTER | 07 |
| IO\$_REWIND | 24 |
| IO\$_REWINDOFF | 22 |
| IO\$_SEARCH | 09 |
| IO\$_SEEK | 02 |
| IO\$_SENSECHAR | 1B |
| IO\$_SENSEMODE | 27 |
| IO\$_SETCHAR | 1A |
| IO\$_SETCLOCK | 37 |

VAX/VMS I/O FUNCTION CODES (CONTINUED)

| | |
|------------------|----|
| IO\$_SETCLOCKP | 05 |
| IO\$_SETMODE | 23 |
| IO\$_SKIPFILE | 25 |
| IO\$_SKIPRECORD | 26 |
| IO\$_SPACEFILE | 02 |
| IO\$_SPACERECORD | 09 |
| IO\$_STARTDATA | 38 |
| IO\$_STARTDATAP | 06 |
| IO\$_STARTMPROC | 02 |
| IO\$_STARTSPNDL | 19 |
| IO\$_STOP | 03 |
| IO\$_TTYREADALL | 3A |
| IO\$_TTYREADPALL | 3B |
| IO\$_UNLOAD | 01 |
| IO\$_VIRTUAL | 3F |
| IO\$_WRITECHECK | 0A |
| IO\$_WRITECHECKH | 18 |
| IO\$_WRITEHEAD | 0D |
| IO\$_WRITELBLK | 20 |
| IO\$_WRITEMARK | 1C |
| IO\$_WRITEOF | 28 |
| IO\$_WRITEPBLK | 0B |
| IO\$_WRITERET | 18 |
| IO\$_WRITETRACKD | 0F |
| IO\$_WRITEVBLK | 30 |
| IO\$_WRITTMKR | 1D |

APPENDAGES TO THE VAX/VMS FUNCTION CODES

Modifiers (prefix to function code) for Proteus Digital Channel
Output Driver (VAX to UYS-1)

- 0 - CIW
- 1 - NIW
- 2 - CW
- 3 - DW

Modifiers for Proteus Digital Channel Input Driver (UYS-1 to VAX)

- 0 - CIW,NIW,CRS,SRL
- 1 - CIW,NIW,DW,CRS,SRL
- 2 - CIW,NIW,CW,CRS,SRL
- 3 - ALL SEQUENCES
- 4 - CRS,SRL

APPENDIX B

Appendix B contains all the possible status codes which may be returned when a VAX/VMS System I/O function occurs.

```

SS$ _ABORT                = 0000002C
SS$ _ACCONFLICT           = 00000800
SS$ _ACCVIO               = 0000000C
SS$ _ACPVAFUL             = 000002FC
SS$ _ALRDYCLOSED         = 000006A9
SS$ _ARTRES               = 00000474
SS$ _ASTFLT               = 0000040C
SS$ _BADATTRIB           = 00000034
SS$ _BADCHKSUM            = 00000808
SS$ _BADESCAPE            = 0000003C
SS$ _BADFILEHDR          = 00000810
SS$ _BADFILENAME         = 00000818
SS$ _BADFILEVER          = 00000820
SS$ _BADIMGHDR           = 00000044
SS$ _BADIRECTORY         = 00000828
SS$ _BADISD               = 00002004
SS$ _BADPARAM             = 00000014
SS$ _BADQFILE            = 000003BC
SS$ _BADQUEUEHDR         = 00000394
SS$ _BADRCT               = 0000216C
SS$ _BADSTACK            = 00000284
SS$ _BADVEC               = 00002064
SS$ _BEGOFFILE            = 00000938
SS$ _BLOCKCNTERR         = 00000940
SS$ _BREAK                = 00000414
SS$ _BUFBYTALI           = 0000030C
SS$ _BUFFEROVF           = 00000601
SS$ _BUFNOTALIGN         = 00000324
SS$ _BUGCHECK             = 000002A4
SS$ _CANCEL               = 00000830
SS$ _CHAINW               = 00000C08
SS$ _CHANINTLK           = 0000004C
SS$ _CLEARED              = 00002104
SS$ _CLIFRCXT             = 00000980
SS$ _CMODSUPR             = 0000041C
SS$ _CMODUSER             = 00000424
SS$ _COMMHARD             = 000020C4
SS$ _COMPAT               = 0000042C
SS$ _CONCEALED            = 00000691
SS$ _CONNECFAIL          = 000020DC
SS$ _CONTINUE             = 00000001
SS$ _CONTROL              = 00000651
SS$ _CONTROLO             = 00000609
SS$ _CONTROLY             = 00000611
SS$ _CREATED              = 00000619
SS$ _CTRLERR              = 00000054
SS$ _CVTUNGRANT           = 0000213C
SS$ _DATACHECK            = 0000005C
SS$ _DATAOVERUN           = 00000838
SS$ _DBGOPCREQ            = 000006A1
SS$ _DEADLOCK             = 00000E0A
SS$ _DEBUG                = 0000046C
SS$ _DECOVF               = 000004A4
SS$ _DEACTIVE             = 000002C4
SS$ _DEVALLOC             = 00000840
SS$ _DEVALRALLOC         = 00000641
SS$ _DEVASSIGN            = 00000848
SS$ _DEVCMDERR            = 0000032C
SS$ _DEVFOREIGN           = 00000064
SS$ _DEVICEFULL           = 00000850

```

```

SS$ _DEVINACT             = 000020D4
SS$ _DEVMOUNT             = 0000006C
SS$ _DEVNOTALLOC         = 00000858
SS$ _DEVNOTDISM          = 000021B4
SS$ _DEVNOTMBX            = 00000074
SS$ _DEVNOTMOUNT         = 0000007C
SS$ _DEVOFFLINE          = 00000084
SS$ _DEVREQERR           = 00000334
SS$ _DGQINCOMP           = 000009C0
SS$ _DIRALLOC            = 000009C8
SS$ _DIRFULL              = 00000860
SS$ _DIRNOTEMPTY         = 00002174
SS$ _DISCONNECT          = 0000204C
SS$ _DRVERR               = 0000008C
SS$ _DUPDSKQUOTA         = 000003DC
SS$ _DUPFILENAME         = 00000868
SS$ _DUPLNAM              = 00000094
SS$ _DUPUNIT              = 000021C4
SS$ _ENDOFFILE           = 00000870
SS$ _ENDOTAPE            = 00000878
SS$ _ENDOFUSRLBL         = 00000970
SS$ _ENDOFVOLUME         = 000009A0
SS$ _EOTIN                = 00000C03
SS$ _EXASTLM              = 00002A04
SS$ _EXBIOLM              = 00002A0C
SS$ _EXBYTLM              = 00002A14
SS$ _EXCPUTIM             = 000020AC
SS$ _EXDEPTH              = 00000E1A
SS$ _EXDIOLM              = 00002A1C
SS$ _EXDISKQUOTA         = 000003EC
SS$ _EXENQLM              = 00002A44
SS$ _EXFILLM              = 00002A24
SS$ _EXGBLPAGFIL         = 00002164
SS$ _EXPGFLQUOTA         = 00002A2C
SS$ _EXPORTQUOTA         = 000003AC
SS$ _EXPRCLM              = 00002A34
SS$ _EXQUOTA              = 0000001C
SS$ _EXQUOTAEND          = 00002AFF
SS$ _EXQUOTASTRT         = 00002A00
SS$ _EXTIDXFILE          = 00000880
SS$ _EXTQELM              = 00002A3C
SS$ _FACILITY             = 00000000
SS$ _FCPREADERR          = 00000888
SS$ _FCPREPSTN           = 00000988
SS$ _FCPREWNDERR         = 00000890
SS$ _FCSPACERR           = 00000898
SS$ _FCPWRTERR           = 000008A0
SS$ _FILACCERR           = 0000009C
SS$ _FILALRACC           = 000000A4
SS$ _FILELOCKED          = 000008A8
SS$ _FILENUMCHK          = 000008B0
SS$ _FILEPURGED          = 00000679
SS$ _FILESEQCHK          = 000008B8
SS$ _FILESTRUCT          = 000008C0
SS$ _FILNOTACC           = 000000AC
SS$ _FILNOTCNTG          = 000002AC
SS$ _FILNOTEXP           = 000000B4

```

\$\$\$ _FLTIDIV = 00000494
 \$\$\$ _FLTIDIV_F = 000004BC
 \$\$\$ _FLTQVF = 0000048C
 \$\$\$ _FLTQVF_F = 000004B4
 \$\$\$ _FLTUND = 0000049C
 \$\$\$ _FLTUND_F = 000004C4
 \$\$\$ _FORCEDERROR = 00002144
 \$\$\$ _FORCEDEXIT = 0000217C
 \$\$\$ _FORMAT = 000000BC
 \$\$\$ _GPTFULL = 000000C4
 \$\$\$ _GSDFULL = 000000CC
 \$\$\$ _HANGUP = 000002CC
 \$\$\$ _HEADERFULL = 000008C8
 \$\$\$ _IDMISMATCH = 000003F4
 \$\$\$ _IDXFILEFULL = 000008D0
 \$\$\$ _ILLBLKNUM = 000000DC
 \$\$\$ _ILLCDTST = 00002154
 \$\$\$ _ILLCNTRFUNC = 000000E4
 \$\$\$ _ILLEFC = 000000EC
 \$\$\$ _ILLIOFUNC = 000000F4
 \$\$\$ _ILLBLAST = 00000968
 \$\$\$ _ILLPAGCNT = 000000FC
 \$\$\$ _ILLSELF = 0000214C
 \$\$\$ _ILLSEQOP = 000002DC
 \$\$\$ _ILLSER = 00000104
 \$\$\$ _ILLUSRLBLRD = 00000958
 \$\$\$ _ILLUSRLBLWT = 00000960
 \$\$\$ _INCOMPAT = 00000699
 \$\$\$ _INCVOLLABEL = 0000010C
 \$\$\$ _INHCHME = 000004D4
 \$\$\$ _INHCHMK = 000004CC
 \$\$\$ _INSFARG = 00000114
 \$\$\$ _INSFBUFDP = 0000033C
 \$\$\$ _INSFCDT = 000021AC
 \$\$\$ _INSFMAPREG = 00000344
 \$\$\$ _INSFMEM = 00000124
 \$\$\$ _INSFRAME = 0000012C
 \$\$\$ _INSFSPTS = 00002044
 \$\$\$ _INSFWSL = 0000011C
 \$\$\$ _INTDIV = 00000484
 \$\$\$ _INTERLOCK = 0000038C
 \$\$\$ _INTQVF = 0000047C
 \$\$\$ _INVLOGIN = 0000209C
 \$\$\$ _IVADDR = 00000134
 \$\$\$ _IVBUFLEN = 0000034C
 \$\$\$ _IVCHAN = 0000013C
 \$\$\$ _IVCHAR = 000020CC
 \$\$\$ _IVCHNLSEC = 0000026C
 \$\$\$ _IVDEVNAM = 00000144
 \$\$\$ _IVGSDNAM = 0000014C
 \$\$\$ _IVLOCKID = 00002124
 \$\$\$ _IVLOGNAM = 00000154
 \$\$\$ _IVLOGTAB = 0000015C
 \$\$\$ _IVLVEC = 0000203C
 \$\$\$ _IVMODE = 00000354
 \$\$\$ _IVPROTECT = 000002F4
 \$\$\$ _IVQUOTAL = 00000164

\$\$\$ _IVSECFLG = 0000016C
 \$\$\$ _IVSECIDCTL = 000002E4
 \$\$\$ _IVSSRQ = 00000174
 \$\$\$ _IVSTSFLG = 0000017C
 \$\$\$ _IVTIME = 00000184
 \$\$\$ _JBCERROR = 0000218C
 \$\$\$ _LCKPAGFUL = 000000D4
 \$\$\$ _LENVIO = 0000018C
 \$\$\$ _LINEABRT = 00000E02
 \$\$\$ _LINKABORT = 000020E4
 \$\$\$ _LINKDISCON = 000020EC
 \$\$\$ _LINKEXIT = 000020F4
 \$\$\$ _LKWSETFUL = 00000194
 \$\$\$ _MBFULL = 000008D8
 \$\$\$ _MBTOOSML = 0000019C
 \$\$\$ _MCHECK = 000002BC
 \$\$\$ _MCNOTVALID = 0000035C
 \$\$\$ _MEDOFL = 000001A4
 \$\$\$ _MSGNOTFND = 00000621
 \$\$\$ _MTLBLELONG = 00000304
 \$\$\$ _MULTRMS = 0000202C
 \$\$\$ _MUSTCLOSEFL = 00000948
 \$\$\$ _NOACNT = 0000284C
 \$\$\$ _NOALLSPOOL = 00002824
 \$\$\$ _NOALTPRI = 0000286C
 \$\$\$ _NOAQB = 00000314
 \$\$\$ _NOBUGCHK = 0000288C
 \$\$\$ _NOBYPASS = 000028EC
 \$\$\$ _NOCMEXEC = 0000280C
 \$\$\$ _NOCMKRNL = 00002804
 \$\$\$ _NODATA = 000001AC
 \$\$\$ _NODETACH = 0000282C
 \$\$\$ _NODEVAVL = 000009B0
 \$\$\$ _NODIAGNOSE = 00002834
 \$\$\$ _NODISKQUOTA = 000003E4
 \$\$\$ _NOEXQUOTA = 0000289C
 \$\$\$ _NOGROUP = 00002844
 \$\$\$ _NOGRPNAM = 0000281C
 \$\$\$ _NOHANDLER = 000008F8
 \$\$\$ _NOHOMEBLK = 000008E0
 \$\$\$ _NOIOCHAN = 000001B4
 \$\$\$ _NOLICENSE = 00002194
 \$\$\$ _NOLINKS = 0000027C
 \$\$\$ _NOLISTENER = 0000215C
 \$\$\$ _NOLOCKID = 00000E12
 \$\$\$ _NOLOGNAM = 000001BC
 \$\$\$ _NOLOG_IO = 0000283C
 \$\$\$ _NOMBX = 00000274
 \$\$\$ _NOMOREFILES = 00000930
 \$\$\$ _NOMOREPROC = 000009A8
 \$\$\$ _NOMOUNT = 0000288C
 \$\$\$ _NONETMBX = 000028A4
 \$\$\$ _NONEXDRV = 000001C4
 \$\$\$ _NONEXPR = 000008E8
 \$\$\$ _NONLOCAL = 000008F0
 \$\$\$ _NOOPER = 00002894
 \$\$\$ _NOP1VA = 00002024

```

SS$_NOPFNMAP      = 000028D4
SS$_NOPHY_IO      = 000028B4
SS$_NOPRIV        = 00000024
SS$_NOPRIVEND     = 000029FF
SS$_NOPRIVSTRT    = 00002800
SS$_NOPRMCEB      = 00002854
SS$_NOPRMGBL      = 000028C4
SS$_NOPRMMBX      = 0000285C
SS$_NOPSWAPM      = 00002864
SS$_NOQFILE       = 000003C4
SS$_NORMAL        = 00000001
SS$_NOSETPRV      = 00002874
SS$_NOSHMBLOCK    = 000003B4
SS$_NOSHMEM       = 000028DC
SS$_NOSHRIMG      = 000021BC
SS$_NOSIGNAL      = 00000900
SS$_NOSLOT        = 0000039C
SS$_NOSOLICIT     = 00000284
SS$_NOSUCHDEV     = 00000908
SS$_NOSUCHFILE    = 00000910
SS$_NOSUCHNODE    = 0000028C
SS$_NOSUCHOBJ     = 000020A4
SS$_NOSUCHSEC     = 00000978
SS$_NOSUCHUSER    = 00002084
SS$_NOSYSGBL      = 000028CC
SS$_NOSYSLCK      = 000028F4
SS$_NOSYSNAM      = 00002814
SS$_NOSYSPRV      = 000028E4
SS$_NOTALLPRIV    = 00000681
SS$_NOTAPEOP      = 00000264
SS$_NOTCREATOR    = 00000384
SS$_NOTFILEDEV    = 000001CC
SS$_NOTINSTALL    = 00002014
SS$_NOTINTBLSZ    = 000001D4
SS$_NOTLABELMT    = 000001DC
SS$_NOTMODIFIED   = 00000659
SS$_NOTMPMBX      = 0000287C
SS$_NOTNETDEV     = 000002EC
SS$_NOTPRINTED    = 00002184
SS$_NOTQUEUED     = 00000988
SS$_NOTRAN        = 00000629
SS$_NOTSQDEV      = 000001E4
SS$_NOTVOLSET     = 00000998
SS$_NOVOLPRO      = 000028AC
SS$_NOWORLD       = 00002884
SS$_NOWRT         = 000003FC
SS$_OPCCUS        = 00000434
SS$_OPCDEC        = 0000043C
SS$_OPINCOMPL     = 000002D4
SS$_OPRABORT      = 000020B4
SS$_OVRDSKQUOTA   = 00000669
SS$_PAGOWNVIO     = 000001EC
SS$_PAGRDERR      = 00000444
SS$_PARITY        = 000001F4
SS$_PARNOTGRANT   = 00002134
SS$_PARTESCAPE    = 000001FC
SS$_PARTMAPPED    = 00000E22

```

```

SS$_PATHLOST      = 000020FC
SS$_PFMBYS        = 00000204
SS$_PLHLDR        = 00000404
SS$_POWERFAIL     = 00000364
SS$_PRIVINSTALL   = 00002054
SS$_PROTINSTALL   = 0000205C
SS$_PROTOCOL      = 00002074
SS$_PSTFULL       = 0000020C
SS$_QFACTIVE      = 000003CC
SS$_QFNOTACT      = 000003D4
SS$_RADRMOD       = 0000044C
SS$_RDELDATA      = 00000661
SS$_REJECT        = 00000294
SS$_RELINK        = 0000200C
SS$_REMOTE        = 00000649
SS$_REMRSRC       = 0000206C
SS$_RESET         = 0000210C
SS$_RESIGNAL      = 00000918
SS$_RESULTOVF     = 00000214
SS$_ROPRAND       = 00000454
SS$_SECTBLFUL     = 0000021C
SS$_SHARTOOBIG    = 0000201C
SS$_SHMGSNOTMAP   = 0000036C
SS$_SHMNOTCNCT    = 0000037C
SS$_SHRIDMISMAT   = 000020BC
SS$_SHUT          = 0000208C
SS$_SSFAIL        = 0000045C
SS$_SUBLOCKS      = 0000212C
SS$_SUBRNG        = 0000044C
SS$_SUPERSEDE     = 00000631
SS$_SUSPENDED     = 000003A4
SS$_SYNCH         = 00000689
SS$_SYSVERDIF     = 00000671
SS$_TAPEPOSLOST   = 00000224
SS$_TBIT          = 00000464
SS$_THIRDPARTY    = 0000207C
SS$_TIMEOUT       = 0000022C
SS$_TOOMANYLNAM   = 00000374
SS$_TOOMANYREDS   = 0000211C
SS$_TOOMANYVER    = 00000990
SS$_TOOMUCHDATA   = 0000029C
SS$_UNASEFC       = 00000234
SS$_UNREACHABLE   = 00002094
SS$_UNSAFE        = 0000023C
SS$_UNSOLICIT     = 00002114
SS$_UNWIND        = 00000920
SS$_UNWINDING     = 00000928
SS$_VASFULL       = 00000244
SS$_VCBROKEN      = 0000219C
SS$_VCCLOSED      = 000021A4
SS$_VECFULL       = 00002034
SS$_VECINUSE      = 0000024C
SS$_VOLINV        = 00000254
SS$_WAITUSRLBL    = 00000950
SS$_WASCLR        = 00000001
SS$_WASECC        = 00000639
SS$_WASSET        = 00000009
SS$_WRITLCK       = 0000025C
SS$_WRONGACP      = 0000031C

```

REFERENCES

1. "ASP FASP User's Manual," Cerquitella, C.V., NADC, 21 July 1983.
2. "ASP Relocating Loader User's Manual," IBM Federal Systems Division, Manassas, Virginia, 30 April 1976.
3. "ASP Service Test Model: SPL Cross Assembler User's Manual," IBM Federal Systems Division, Manassas, Virginia, 27 April 1977.
4. "FASP User's Guide," Structured Technology Corporation, New London, Connecticut, 14 May 1984.
5. "Principles of Operation for the Proteus Control Processor and Digital I/O Commands," IBM Federal Systems Division, Manassas, Virginia, 12 January 1976.
6. "Support Software for the Proteus Digital Channel/NTDS/UNIBUS Interface," Vernaglia, J.E., NUSC Technical Memorandum, 18 December 1978.
7. "User's Guide for the Test Bed Command Language," Bernecky, W.R., NUSC Technical Memorandum, 27 May 1977.
8. "VAX/VMS I/O User's Guide," Digital Equipment Corporation, Maynard, Massachusetts, January 1981.
9. "VAX/VMS System Services Reference Manual," Digital Equipment Corporation, Maynard, Massachusetts, March 1980.

TM No. 851005

PROCEDURE FOR USING THE FASP

Barbara Bower Welles

Electronic Engineering and Computer Applications Division

Surface Ship Sonar Department

TM No. 851005

8 January 1985

UNCLASSIFIED

DISTRIBUTION LIST

Internal

Codes

038

33

331

33B (J. Vernaglia, R. Wimberger, M. Simard)

33B3 (C. Nawrocki, M. Burt)

33B312 (N. Straatveit)

3311 (K. Springer)

3312 (R. Rubega, C. Becker)

3313 (M. Cloutier, S. Thorp, E. Marvin, R. Kosman, V. Kondratenko, S. Park, M. Maugle, E. Maiser, G. Panko)

3314 (C. Carter, R. Lai, D. McKay, S. Sutherland, D. Lerro, D. Sheldon, K. Scarbrough, P. Stahl)

3314 B. Welles (5 copies)

70

Total 35